Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.061948

ARTICLE

Integrating Attention Mechanisms in YOLOv8 for Improved Fall Detection Performance

Nizar Zaghden[1], **Emad Ibrahim**[2], **Mukaram Safaldin**[2,*] **and Mahmoud Mejdoub**[3]

[1]Higher School of Business of Sfax, University of Sfax, Sfax, 3018, Tunisia
[2]National School of Electronics and Telecommunications of Sfax, University of Sfax, Sfax, 3029, Tunisia
[3]Faculty of Sciences of Sfax, University of Sfax, Sfax, 3018, Tunisia

[*] Corresponding Author: Mukaram Safaldin. Email: mukarramalqadiry@gmail.com

Received: 06 December 2024; Accepted: 24 February 2025; Published: 26 March 2025

ABSTRACT: The increasing elderly population has heightened the need for accurate and reliable fall detection systems, as falls can lead to severe health complications. Existing systems often suffer from high false positive and false negative rates due to insufficient training data and suboptimal detection techniques. This study introduces an advanced fall detection model integrating YOLOv8, Faster R-CNN, and Generative Adversarial Networks (GANs) to enhance accuracy and robustness. A modified YOLOv8 architecture serves as the core, utilizing spatial attention mechanisms to improve critical image regions' detection. Faster R-CNN is employed for fine-grained human posture analysis, while GANs generate synthetic fall scenarios to expand and diversify the training dataset. Experimental evaluations on the DiverseFALL10500 and CAUCAFall datasets demonstrate that the proposed model significantly outperforms state-of-the-art methods. The model achieves a mean Average Precision (mAP) of 0.9507 on DiverseFALL10500 and 0.996 on CAUCAFall, surpassing conventional YOLO and R-CNN-based models. Precision and recall metrics also indicate superior detection performance, with a recall of 0.929 on DiverseFALL10500 and 0.9993 on CAUCAFall, ensuring minimal false negatives. Real-time deployment tests on the Xilinx Kria™ K26 System-on-Module confirm an average inference time of 43ms per frame, making it suitable for real-time monitoring applications. These results establish the proposed R-CNN_GAN_YOLOv8 model as a benchmark in fall detection, offering a reliable and efficient solution for healthcare applications. By integrating attention mechanisms and GAN-based data augmentation, this approach significantly enhances detection accuracy while reducing false alarms, improving safety for elderly individuals and high-risk environments.

KEYWORDS: DiverseFALL10500; CAUCAFall; faster region-based conventional neural network (Faster RCNN); generative adversarial networks (GANs); human fall; YOLO

1 Introduction

In recent decades, advancements in quality of life have led to longer lifespans, resulting in a rapidly growing elderly population. Elderly healthcare has become a global concern, drawing significant attention from healthcare practitioners, engineers, and the healthcare industry. According to the United Nations (UN), the population aged 60 years and older is projected to increase from 900 million in 2015 to 2 billion by 2050 [1,2]. Among the various health challenges faced by the elderly, slip-and-fall accidents are a leading cause of fatalities in old age [3,4]. Additionally, most falls resulting in injuries occur primarily at home. With increasingly fast-paced and demanding lifestyles, many families worldwide cannot provide constant care for elderly relatives. This necessitates the development of intelligent fall detection systems to assist elderly individuals at risk of falls when caregivers are unavailable [5–7]. Moreover, fall detection technology can also be deployed in other high-risk environments to prevent fatalities caused by slipping and falling [8,9]. Consequently, implementing a reliable real-time human fall detection system is crucial.

Artificial Intelligence (AI) has emerged as a transformative technology in recent years, offering wide-ranging applications in the healthcare sector [10–12]. Fig. 1 presents an example of an image-processing-based fall detection system. This setup employs multiple cameras strategically positioned around a room to capture various angles and viewpoints, enabling comprehensive monitoring of activities within the space. The system focuses specifically on detecting falls in real time. As shown in Fig. 1, the cameras are configured to monitor individuals, such as a person with a cane, identifying fall incidents through advanced image processing techniques. Such systems are particularly beneficial for enhancing the safety of elderly individuals or those with mobility challenges by providing timely alerts and ensuring rapid responses during a fall.

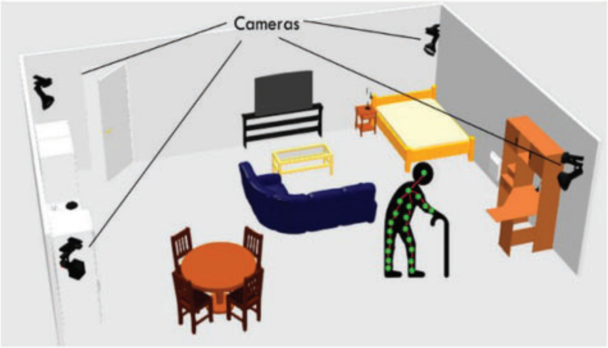


Figure 1: Example of fall detection system

Cyber-physical system monitoring allows us to recognize, understand, and attempt to avoid events and accidents whenever possible to improve system reliability and robustness. Regular monitoring of crowded places is helpful for public safety. The manual control of these overcrowded places is labour-intensive. The automatic video system is the best for human detection [13,10]. Fall detection systems based on different types of sensors are beneficial for improving the security of construction industries [14]. Various video sequence box-like texture information and skeleton information are used to detect a human falling object [15,16]. The availability of video data has long been the key to the success of AI models.

A human body can be detected differently across the light environment; to overcome this issue, researchers have utilized different features such as the shape, speed, and vibration of some objects to identify humans. The classification of speed, shape, and texture can improve the precision of the robust moving object classifier in the alternative lighting system. Addressing the data sparsity problem due to lacking information on falls in previous research by introducing an improved model of human fall detection. Therefore, in this paper, we propose an enhanced model of (Region-based Convolutional Neural Network) R-CNN, YOLO v8 with a GAN to increase the effectiveness of fall detection.

1.1 YOLO (You Only Look Once) Object Detection

The YOLO object detection method, developed by Joseph Redmon et al. in [17], is a real-time object detection. It removes the various bottlenecks present in the other detection pipelines. YOLO detects bounding boxes and classifies them simultaneously in a uniquely creative manner. Fig. 2 illustrates that the YOLO architecture is a convolutional neural network for real-time object detection. It processes an input image through convolutional layers with various kernel sizes, each followed by max-pooling layers to reduce the spatial dimensions and extract features. The architecture begins with a 7 × 7 convolution followed by a 2 × 2 max-pooling layer, then progresses through several blocks of 3 × 3 convolutions interspersed with 1 × 1 convolutions to increase the depth and complexity of the feature maps [14,18–20].

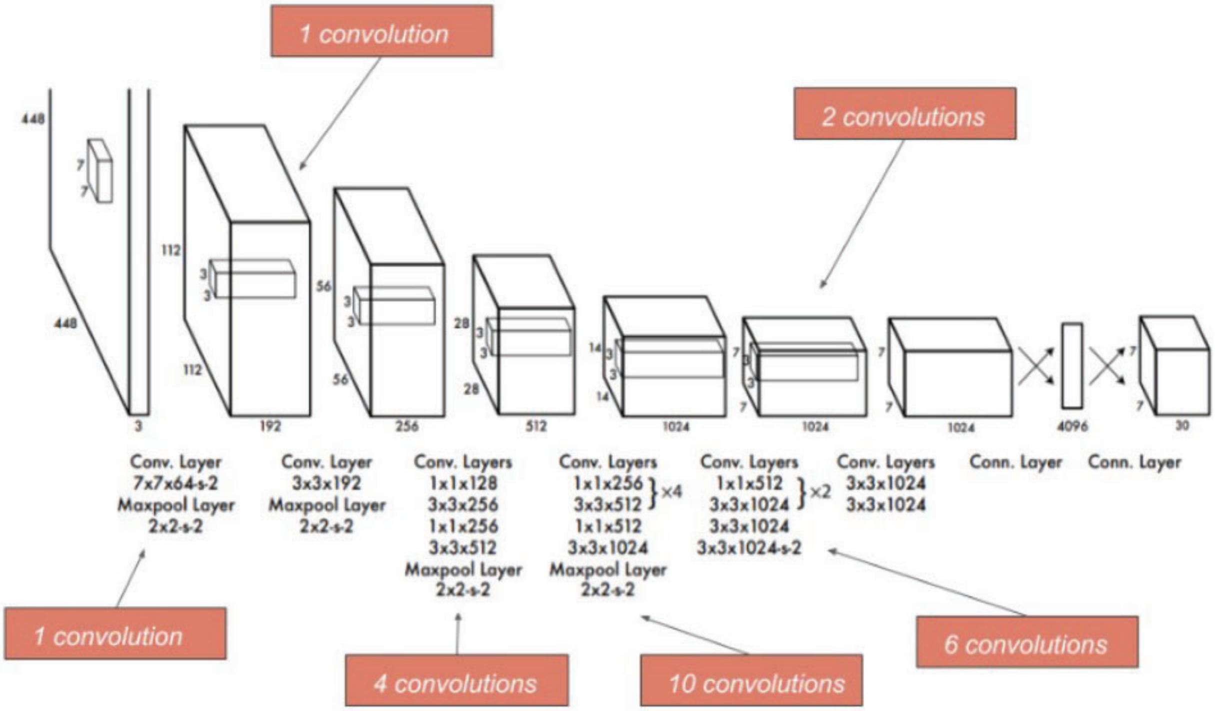
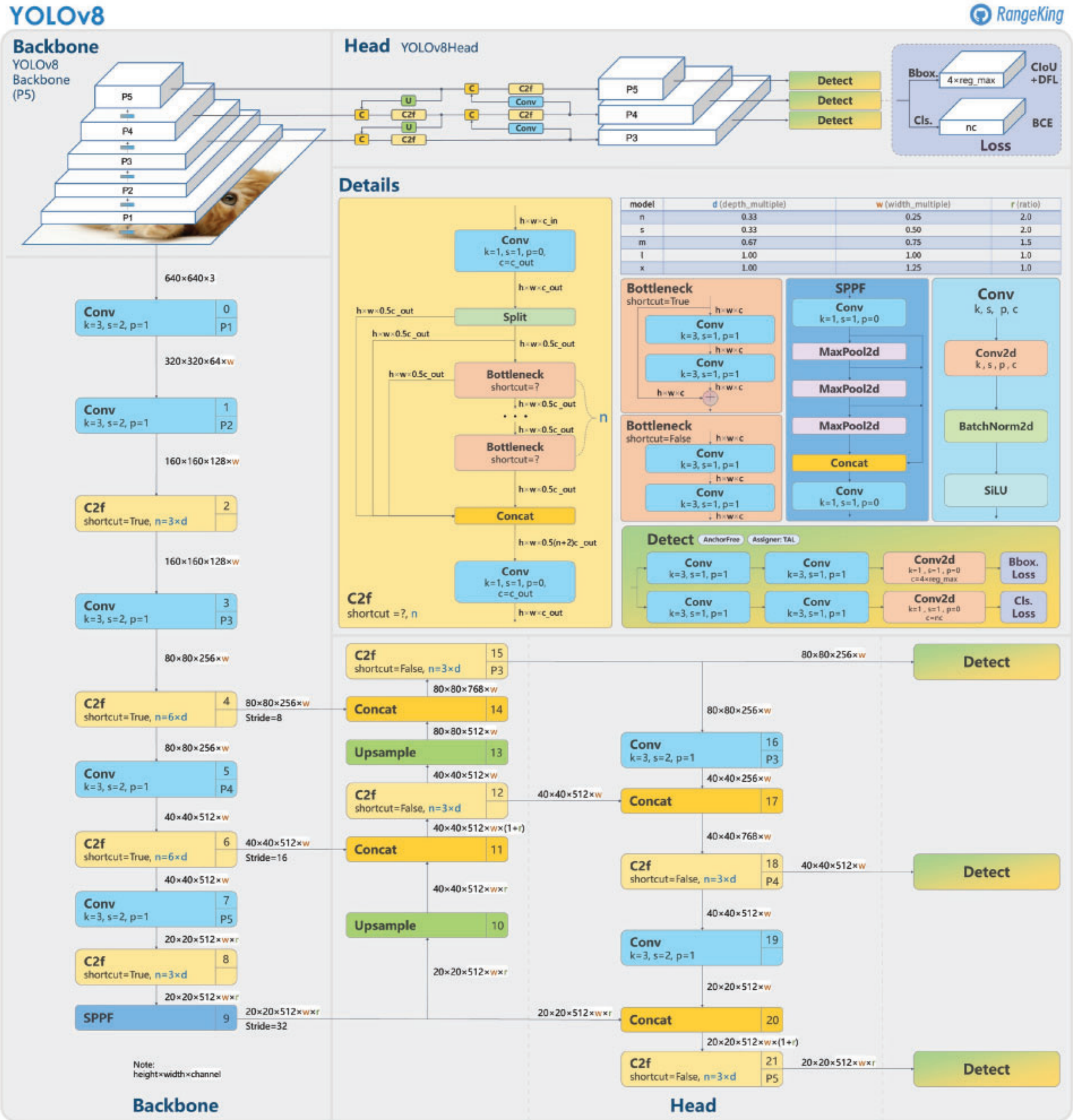


Figure 2: Architecture of original YOLO

The network then transitions through 4, 10, and 6 convolutions in subsequent layers, using 1 × 1 and 3 × 3 filters, with periodic max-pooling to downsample the feature maps. The final layers consist of fully connected layers that output a tensor predicting bounding boxes and class probabilities for detected objects. This design enables YOLO to perform detection and classification in a single forward pass, making it highly efficient for real-time applications.

YOLO divides the input image into an $S \times S$ grid. If the object's centre falls into a grid cell, that cell is responsible for detecting that object. Each grid cell predicts B boundary boxes and confidence scores for those boxes. The confidence score shows how the box 'resembles' an object and how much it overlaps with the ground truth. Also, each grid cell predicts C conditional class probabilities. These class probabilities are conditioned on the grid cell containing an object and decide which class(es) the object belongs to. The final output is a $S \times S \times (B \times 5 + C)$ [20].

If the final detected box crosses the mid-point threshold of 0.5, the model captures the object as a valid detection. This means the model considers the object within the bounding box, provided that the confidence score and class probability also support this detection. The mid-point threshold of 0.5 indicates that more than half of the object lies within the detected bounding box, ensuring that the object is accurately captured. Suppose the detection fails to meet this threshold. In that case, it may be discarded as a false positive or an incomplete detection, thus maintaining the reliability of the YOLO model in identifying objects in the image. Fig. 3 illustrates the architecture of YOLOv8; it builds on its predecessors with enhancements in both the backbone and the head of the network, designed for improved object detection performance [21].



model	d (depth_multiple)	w (width_multiple)	r (ratio)
n	0.33	0.25	2.0
s	0.33	0.50	2.0
m	0.67	0.75	1.5
l	1.00	1.00	1.0
x	1.00	1.25	1.0

Figure 3: Architecture of YOLOv8

The backbone consists of convolutional layers and Cross Stage Partial (CSP) with 2 convolutions and Feature enrichment (C2f) modules that process the input image at multiple scales (P1 to P5), extracting features at different resolutions. Each C2F module incorporates shortcut connections for better gradient flow and feature reuse, the backbone outputs feature maps at various scales, which are then passed to the head. The head utilizes these multi-scale features through convolutional layers, concatenations, and upsampling operations to refine the detection predictions. It includes specific detection layers for predicting bounding boxes, objectness scores, and class probabilities. The architecture emphasizes efficiency and accuracy by combining deep feature extraction, efficient scaling, and precise detection mechanisms, suitable for various real-time object detection applications [20–22].

1.2 Generative Adversarial Networks (GANs)

The main two components of the GAN are the discriminator and the generator. Both will be trained using different loss functions. A GAN aims to generate data from randomly selected data points [23,24]. The generator output will be validated or rejected by comparing the result from the output of the discriminator. The generator will try to generate the output data for the randomly selected point and send the result to the discriminator. If the discriminator's classification result is wrong, the generator updates the weights following its architecture using generator loss. The discriminator will classify the input generated by the generator and the actual output data. If the discriminator's classification of the input is wrong, then the loss will be transmitted back to the generator using the architecture.

In 2014, Ian Goodfellow was the first person to introduce the GAN [23]. The GAN addresses many problems and is broadly accepted by the community due to its application in computer vision problems, recommendation systems, anomaly detection, and machine translations. From the original version of 2014, there are also some promising works with a high impact on the state of the art in computer vision. Another interesting GAN discussed in the literature is C-GAN [25]. In the C-GAN, there is also a condition vector that connects the generator and the discriminator to the image creation portion of the GAN, as shown in the architecture in Fig. 4. GAN has many improvements from the original idea for stability and conditional ones. The training in GAN architecture can be divided into two main steps.

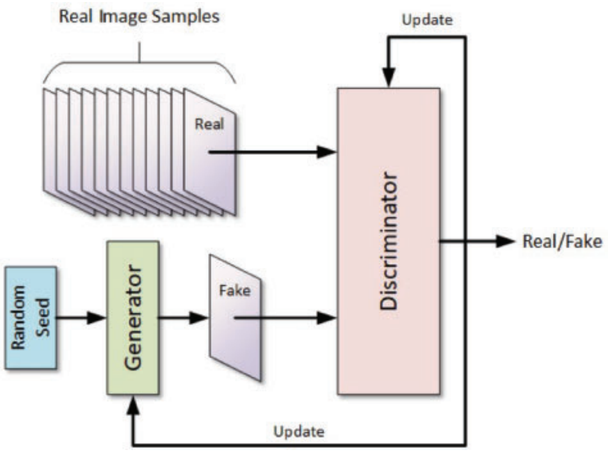


Figure 4: Architecture of GAN

The architecture, as depicted in Figure of a GAN 4, comprises two neural networks: one is called the generator, while the other is called the discriminator; it is trained simultaneously. Input to the generator may be taken from any random seed, and the generated fake samples and actual samples are provided as input to the discriminator. Hence, this discriminator's paramount task would be to tell whether the samples are fake or real and, thus, let the generator know through particular feedback. Therefore, according to the result obtained by this discriminator after training a network, an update regarding the generator and discriminator may be carried out accordingly [26]. The generator aims to improve its ability to create real samples; conversely, the discriminator agent strives to strengthen its capability to detect real and fake cases correctly. The stopping criteria for this process progresses when the generator produces highly realistic data that the discriminator agent can not detect as a fake sample [27].

RCNN represents a significant advancement in the field of object detection. Initially introduced by Ross Girshick [28]. RCNN revolutionized how objects are detected in images by combining region proposals with Convolutional Neural Networks (CNNs) [29]. The core idea behind RCNN is first to generate a set of region proposals that potentially contain objects. These proposals are then fed into a CNN to extract features classified into different object categories. This two-stage process allows R-CNN to focus computational resources on relevant parts of the image, leading to more accurate detection results than earlier methods.

One of the key strengths of RCNN lies in its ability to leverage pre-trained CNN models, such as AlexNet or VGG, for feature extraction [28]. This transfer learning approach enables RCNN to utilize powerful, pre-learned feature representations, significantly enhancing its object detection capabilities. By applying these deep neural networks to the proposed regions, RCNN can capture intricate details and spatial hierarchies within the image, allowing it to differentiate between objects even in cluttered and complex scenes. Moreover, RCNN's modularity means that improvements in CNN architectures directly benefit the performance of the object detection pipeline.

Despite its effectiveness, the original R-CNN had some limitations, particularly regarding computational efficiency and speed [14,30,31]. The process of generating region proposals, running a CNN on each proposal, and then classifying them was computationally intensive and slow, making R-CNN impractical for real-time applications. This led to faster variants, such as Fast RCNN [28] and Faster RCNN [29]. Fast R-CNN improved the efficiency by integrating the region proposal and feature extraction steps. At the same time, Faster RCNN introduced a Region Proposal Network (RPN) that shares convolutional layers with the detection network, significantly speeding up the process [32]. These advancements have made R-CNN and its variants some of the most powerful and widely used techniques for object detection in modern computer vision applications.

Fig. 5 depicts the Faster R-CNN architecture for object detection begins with an input image, from which Regions of Interest (ROIs) are identified using techniques like selective search. The entire image is then passed through a CNN to extract a feature map. Each ROI is projected onto this feature map, and ROI pooling is applied to produce a fixed-size ROI feature vector. This vector is fed into a Fully Connected (FC) layer, which outputs two branches: one for class probability, determining the object class within the ROI, and another for bounding box regression, refining the coordinates of the detected object. This process allows R-CNN to detect and localize objects within an image effectively.

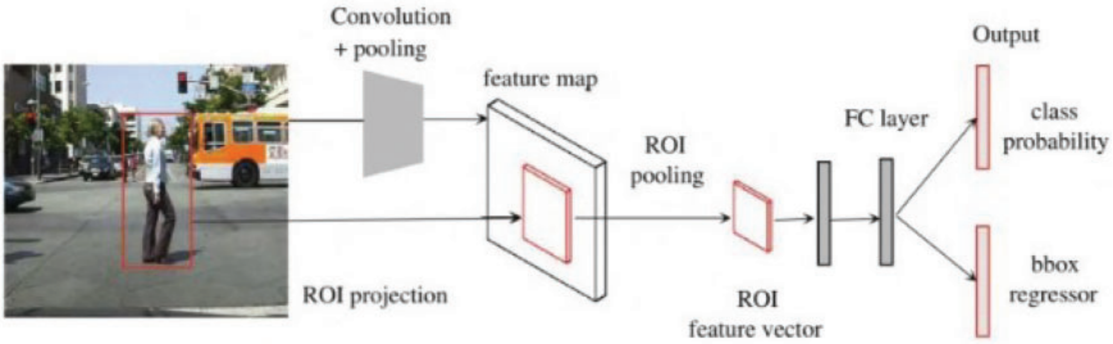


Figure 5: Architecture of faster R-CNN

1.3 Research Significance

Human fall detection is a recurring research topic in the field of computer vision and image recognition [33]. Human or action detection in video is the baseline for human fall detection. Accidents faced by elderly people, inside or outside the house, are increasing, creating an urgency to provide new fall detection solutions. According to World Health Organization (WHO) global statistics, at least 684,000 fatal falls occur each year [34], occupying a significant fraction of all non-fatal injury hospital admissions. Hence, there is a surge in recent trends where researchers have attempted various techniques and employed modern tools to perform research about fall detection in applications using advanced smart home networks [6], personal wear [35], or to keep the surveillance around the environment [36]. However, most existing algorithms and models are shallow and can only capture partial information about the original image, adversely affecting prediction accuracy. Some artifacts also degrade the image quality and training effectiveness.

In addition, most networks are not robust, as they perform unsatisfactorily for images with different textures and low contrast. We enhance the human or action detection process by developing a deeper model to overcome these shortages. To handle the second issue, i.e., the problem caused by the artifacts, we employ GANs to generate realistic synthetic data, which can augment the training dataset. By creating diverse and artifact-free images representing various fall scenarios, GANs help build robust models less sensitive to artifacts. Also, using R-CNN plays a crucial role, especially in feature extraction for the benchmark dataset. R-CNN precisely identifies and localizes humans in various poses, including falling, by generating region proposals and classifying them accurately.

1.4 Literature Limitations

The proposed model makes significant contributions to fall detection by addressing key limitations observed in the existing literature, which hinder current systems' performance, reliability, and applicability. These limitations include:

1. Performance in real-world scenarios requires improvement: By integrating YOLOv8, Faster R-CNN, and GANs, the proposed model improves fall detection performance in diverse and uncontrolled environments, including varying lighting conditions, occlusions, and participant diversity.
2. Insufficient robust training data: GANs for synthetic data generation enrich the training dataset, addressing the challenge of insufficient and non-diverse fall scenario data.
3. High false positive and negative rates: The model's advanced architecture, including attention mechanisms and detailed feature extraction, reduces false alarms and missed detections, improving reliability.
4. Shallow models: The enhanced architecture captures comprehensive image features, addressing the limitations of shallow models in accurately detecting falls.
5. Challenges in low-contrast and texture variations: Incorporating attention mechanisms and advanced feature extraction ensures robust performance across varying image textures and low-contrast conditions.

1.5 Paper Contributions

The contributions of this paper are as follows:

1. A modified architecture of YOLOv8 is proposed for human fall detection, optimized to detect fall events with high accuracy and efficiency.
2. Integration of GANs to augment the dataset with realistic fall scenarios, addressing the challenge of insufficient and non-diverse training data.

3. Utilization of faster R-CNN for detailed feature extraction on annotated bounding boxes, enabling precise identification and localization of human falls.
4. Implementation of advanced attention mechanisms:
    - Channel attention in faster R-CNN: Enhances the model's focus on the most informative features, improving detection accuracy.
    - Squeeze-and-excitation in GAN: Improves the generation of realistic fall scenarios by adaptively recalibrating feature responses.
    - Spatial attention in YOLOv8: Enhances the detection of relevant objects in the spatial domain, improving overall performance in fall detection.
5. Development of a state-of-the-art fall detection system designed to enhance the safety of older adults and short-term care personnel in biometric and monitoring systems.

1.6 Paper Organization

The remainder of this paper is organized as follows: Section 2 reviews the related work on fall detection systems and the advancements in deep learning techniques relevant to this study. Section 3 details the methodology, including the architecture of the proposed R-CNN_GAN_YOLOv8 model and the data augmentation strategies employed. Section 4 presents the experimental setup and evaluation metrics used to assess the model's performance; also it discusses the results and compares the proposed model with existing state-of-the-art methods. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2 Literature Review

Apart from ambient systems, the person can be monitored through vision-based systems to detect falls, which can activate emergency alerts or provide immediate help. The current research on various vision-based systems relies on dissimilar types of video cameras for real-time monitoring. They usually use depth [37] or RGB cameras [38], such as the Raspberry Pi and indoor video surveillance cameras [39], for image acquisition to detect falls. Depth Camera computes the 3D information with just one camera and works very well under dark conditions. RGB cameras are deployed everywhere, from low-profile internet cameras to high-end surveillance cameras, which cannot capture 3D information or work under poor lighting conditions. That is mitigated by having multiple RGB cameras and more infrared sensors similar to smartphone cameras. In both varieties, cameras can detect subjects in vision-based fall detection systems.

Generally, the approaches in vision-based systems share most of their procedure: (i) image preprocessing, (ii) background subtraction or foreground segmentation, (iii) feature extraction and (iv) event recognition [40]. What differs for many solutions in this class of approaches is the type of features identified and extracted within the third step of processing. Based on a feature extracted, it is possible to distinguish between four general types of techniques: (i) monitoring the change in the shape, (ii) analysis of the postures, (iii) key-point tracking, and (iv) detecting inactivity [40]. The shape change method typically approximates the subject with a simple shape [41]. This method requires less computational power and is easy to model; hence, it is popular in real-time vision systems. For example, in systems involving Raspberry Pi cameras and home surveillance cameras, the human subject is often approximated by an ellipse, with a minimal rectangle covering it. The aspect ratio of this rectangle is monitored in every frame and compared to a threshold for fall detection.

In one approach, C-motion, which describes the velocity of the subjects, is combined with shape changes to detect falls [41]. Initially, large motions like falls are detected using C-motion. The system then analyzes the orientation and proportion of the subject shape to determine their status. It then checks for a lack of motion and timing how long the subject remains on the ground. Another technique estimates the height-width ratio and the distance between the mid-centre and top-centre of the approximated rectangle to detect falls using a threshold [42]. Similarly, an SVM detects the ellipse shape, head position, and vertical and horizontal projection histograms to identify the subject's activity [43]. Another study approximates the subject as a voxel shape using multiple video cameras, then classifies the subject's status as upright, on the ground, or in between based on the voxel height using fuzzy logic [44]. The main drawback of the shape change method is that it sacrifices accuracy by approximating the subject with simple geometric shapes to achieve faster and smaller calculations.

In contrast, the posture figuring method provides more accurate information by tracking the subject's joints or outlining their body contour. For example, after background subtraction, a Kalman filter with OpenCV is applied to monitor the person through point detection in areas of interest that can even detect tiny movements. The k-NN algorithm applied to this system predicts falls with an accuracy of 96.9%. Another approach employs three independently used depth cameras to analyze the body contour as the foreground feature, labelling events as falls or non-falls using a majority voting technique, reaching an accuracy of 96.5% [40]. Although this approach is accurate, it is computationally intensive. Another approach is to analyze the contour of the subject's body and set a threshold line to differentiate the frames [45]; if the subject falls below the threshold, it is determined that a fall has occurred. Although the methods of posture figuring are pretty accurate, they require considerable computational power and a large amount of training data. They may not provide the speed necessary for real-time applications, making the shape change method more popular despite slightly reducing accuracy.

Key-point tracking is a compromise derived from posture figuring to reduce high computational expenses [46]. This typically projects the posture of the subject but checks just a few key feature points instead of all the pixels. One of the best examples of solid fall detection using depth cameras involves tracking human body parts by extracting the 3D joints of the body and narrowing them down to the head and hip, visible ones. This approach is relatively efficient: the joint extraction takes only a few milliseconds, and the camera operates at 30 fps. This system then takes this head joint distance trajectory as input into an SVM, allowing it to achieve an accuracy of 97.6%. Poonsri et al. proposed an enhanced algorithm for fall detection using continuous frame voting. The paper, at the beginning, does background subtraction through the MoG in an attempt to detect the human subject's contour and track its centroid. The events in every frame are classified; besides, the prediction accuracy goes up by 91.38% with the introduction of continuous frame voting against only 86.1% initially expected. Another is by Su et al. [47], which is based on the depth camera to analyze features like the centre of gravity, using a neural network for classification. The highest accuracy achieved is 98.15% using the MLP classifier. Tracking key points focuses on less occluded parts, like the head. It reduces computation costs and avoids occlusion problems but usually loses some information, yielding false alarms. Quick sitting down in this manner is misjudged for a fall. Generally, the low computing resource method of inactivity detection raises a high rate of false alarms, which require extended periods of inactivity, which is relatively dangerous and, therefore, usually combined with other methods for reliability.

Timely fall detection is necessary for the proper medical response to prevent severe physical injuries such as hip fractures and concussions, as well as mitigating social losses, which may include the loss of independence, reduced quality of life, and social isolation. It gives timely medical intervention and thus minimizes the risk of grave injuries. Various methodologies have been developed for the fall detection of the elderly. Recently, IoT and AI, along with DL and ML methods, have been used in healthcare to automate the diagnosis of abnormal and diseased conditions. Durga & Ferni propose a new deep convolutional neural network model based on Inception for fall detection and classification named INDCNN-FDC [12]. Further, Maray et al. [48] have proposed and implemented a SmartFall system on a commodity smartwatch; nine elderly people followed the tracking. The proposed system is wearable and well-accepted by these nine elderly persons, though it suffers from two key limitations. The first impossibility is gathering considerable amounts of personalized data for training. The model issues too many false positives when used in a real-world environment for fall detection and trained with less data. The second limitation is the problem of model drift: A model trained with data from one device will generalize poorly on data from another device.

Recent advancements have focused on developing portable and efficient fall detection systems to address this issue. One such approach, in [49], utilizes a vision-based fall prediction and detection system powered by a Human Pose Estimation (HPE) model implemented on the Xilinx® Kira™ K26 System-on-Module (SOM). This system incorporates an Intel® RealSense™ D455 depth camera connected to a KV260 platform to enable real-time processing at 60 frames per second. The proposed solution employs a three-stage pipeline architecture comprising quantized YOLOX, Anchor-to-Joint (A2J), and a convolutional neural network (CNN) fall detection model.

The YOLOX model processes RGB frames (640 x 640 x 3) to generate bounding boxes for humans, subsequently discarding RGB data to prioritize privacy, achieving an accuracy of 74% with a 43 ms inference time. The A2J model analyzes depth frames (288 × 288) to produce 15 joint key points per detected individual, with an accuracy of 76% and an average inference time of 49 ms. Finally, the CNN-based fall detection model predicts fall activity by analyzing joint coordinates (x, y, z) across frames, achieving a quantized accuracy of 72.86% with an average inference time of 1.17 ms. The system displays the fall classification output on an LCD, demonstrating its feasibility as a low-power, real-time solution for elderly fall detection. This approach highlights the potential of combining edge computing devices with advanced machine learning models for efficient and privacy-conscious fall detection systems.

As the above literature survey shows, environmental sensors usually hang on wearable devices or devices embedded in video to a patient for fall detection systems conventionally. However, all these wearable devices may inconvenience most of the users due to clumsiness and other disadvantages of the application. To that regard, Inturi et al. [11] propose a vision based mechanism where key indicators of human joint point analysis will form the fulcrum of a fall event. They use the AlphaPose pre-trained network that extracts key points representative of subject joints for further analysis with CNN to make sense of the spatial relationship between the key points. The approach captures long-term dependencies with long short-term memory. Their developed system can detect five kinds of falls and six daily activities. The outcomes were validated on the UPFALL detection dataset, and commendable results were attained compared to state-of-the-art methods. Comparison with OpenPose for keypoint detection shows that AlphaPose always provides better results.

The key evidence involves bridging crucial lacunas in previous fall detection systems, among them the high false favourable return rates ordinary in many conventional models. This paper, therefore, discusses how these schemes can be enhanced with a modified YOLOv8 architecture for more objective and precise object detection performance. Besides, R-CNN is employed for detailed feature extraction on the annotated bounding boxes, enabling the thorough analysis of human posture and movement. The combination significantly enhances the reliability of fall detection, making it suitable for real-world environments, especially for continuous monitoring of the elderly. Another critical gap this research addresses is the limitation of training datasets in representing diverse fall scenarios. Traditional datasets often lack sufficient variety, leading to poor model performance in detecting falls under different conditions. To overcome this, the research integrates GANs trained with real images of humans in walking and falling positions. This approach augments the dataset, providing a broader range of training examples and improving the model's ability to generalize across various fall scenarios. Moreover, by incorporating an attention mechanism (Channel Attention) into the YOLOv8 model, the research further enhances its detection capabilities. The comparative performance analysis with other state-of-the-art models aims to validate these enhancements, ensuring that the proposed fall detection system meets and exceeds current standards in accuracy and reliability.

3 Methodology

This paper details an experimental study of fall detection using a public dataset and an improved deep-learning architecture, as depicted in Fig. 6. The datasets used in this paper are CAUCAFALL [50] and DiverseFALL10500 [51]. CAUCAFall is a database designed to recognize human falls, distinguished by its creation in uncontrolled home environments. It features various conditions, including occlusions, changes in lighting (natural, artificial, and night), diverse participant clothing, background movements, different floor and room textures, varied fall angles, multiple distances from the camera to the fall, and participants of various ages, weights, heights, and dominant legs. This diversity significantly advances fall recognition research. Additionally, CAUCAFall uniquely includes segmentation labels for each image, enabling the implementation of human fall recognition methods using YOLO detectors. CAUCAFall provides this dataset for human fall-detection-related diseases; this dataset is open-source. The dataset contains subject falls (50%) and activities of daily life (ADLs, 50%) [50].

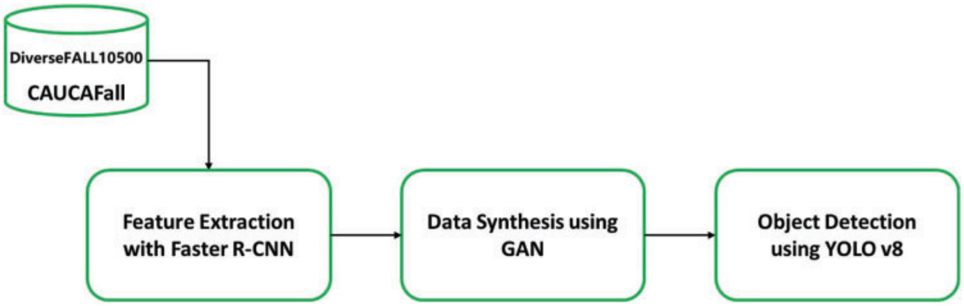

DiverseFALL10500
CAUCAFall
Feature Extraction with Faster R-CNN
Data Synthesis using GAN
Object Detection using YOLO v8

Figure 6: Architecture of the proposed model

It also contains both RGB and depth frames. The dataset consists of two subsets. The first contains six different scenarios. Around 12,600 fall-related activities and 17,274 ADL activities. The second includes two scenarios with 1000 fall-related activities and 1,200 ADL activities. The data can be split into development and evaluation achieves. This dataset is crucial because we can evaluate our trained model in different scenarios. We extract files operating in one-third of the original dataset containing 50% of the falls and 50% of the ADLs. It is essential to use only a certain percentage from our dataset because, as we use an unbalanced dataset, this makes our model train and test equally as if it were fine-tuned in the subset, increasing the robustness of our solution. The CAUCAFall dataset was split into subsets, and only one-third of the data was used for training to achieve multiple objectives. This approach ensures balanced class representation during training and evaluation. The dataset contains an equal proportion of fall-related activities (50%) and activities of daily life (50%). Using a subset with this balanced composition for training, the model is exposed to an equal representation of both classes, avoiding bias toward either falls or non-fall activities and ensuring a fair evaluation process. Limiting the training data to one-third of the dataset was also critical for improving the model's generalization capabilities. Using a smaller portion for training mitigates the risk of overfitting, where the model might become too specific to the training data and fail to perform well on unseen examples. With the remaining two-thirds reserved for validation and testing, the model is evaluated on a broader range of scenarios, ensuring its robustness across diverse and unseen conditions.

Moreover, this subset-based approach enhances the model's robustness to variations in the dataset. The CAUCAFall dataset includes diverse scenarios, such as changes in lighting conditions, participant diversity, and various fall angles. Training on a controlled subset of these variations ensures the model learns to handle a wide range of scenarios without relying on specific data patterns. This careful selection improves the model's generalisation and adaptation to real-world conditions. Additionally, this methodology promotes reproducibility and transparency in research. Using a defined proportion of the dataset, other researchers can easily replicate the experimental setup and validate the findings. This not only supports reproducibility but also enables meaningful comparisons with future studies. Lastly, using a smaller training set reduces computational resource requirements, which is particularly important given the complexity of the proposed model that integrates YOLOv8, Faster R-CNN, and GAN-based augmentation.

On the other hand, the DiverseFALL10500 dataset [51] is a comprehensive and diverse collection of fall detection scenarios specifically developed to address the limitations of existing datasets. It features high variability in visual appearances, fall angles, and environmental conditions, including natural, artificial, and night lighting. The dataset incorporates various age groups, clothing and attire variations, and diverse backgrounds, making it highly representative of real-world conditions. These enhancements ensure that the dataset provides a robust foundation for training and evaluating advanced fall detection models, significantly improving their performance, generalizability, and reliability in complex and uncontrolled environments.

In the preprocessing stage, all instances of lying-down actions in the dataset were explicitly labelled as no-fall cases. This step ensures that the model learns to classify lying-down actions as intentional and controlled movements rather than accidental falls. This labelling is critical to minimizing false positives during the classification phase, particularly in scenarios where individuals might lie down deliberately.

Algorithm 1 outlines the end-to-end pipeline of the proposed system, including data preprocessing, pose estimation, and fall classification. In Step 5, Pose Estimation is performed using a Human Pose Estimation (HPE) model to extract keypoints (*x*, *y*, *z*) of human joints. These keypoints are then processed by a CNN-based fall detection model to classify the activity as either a fall or a non-fall based on the dynamic changes in keypoint positions.

Algorithm 1: Proposed detection approach methodology

```
Step 1: Data Collection and Preparation
Step 2: Feature Extraction with Faster R-CNN
    2.1 Faster R-CNN Integration:
        (a) Use Faster R-CNN to perform detailed feature extraction on detected bounding boxes.
        (b) Integrate Channel Attention:
            i. Apply Channel Attention modules after each convolutional layer in Faster R-CNN to enhance
               feature maps by focusing on the most informative channels.
    2.2 Key Point Detection:
        (a) Apply a pose estimation model (e.g., OpenPose) to the Faster R-CNN output for key point
            (joint) extraction.
Step 3: GANs for Data Synthesis and Enhancement
    3.1 Synthetic Data Generation:
        (a) Use GANs to generate synthetic sequences of falls to augment the training dataset.
        (b) Integrate Squeeze-and-Excitation:
            i. Apply Squeeze-and-Excitation blocks in the GAN architecture to enhance realism in generated
               fall scenarios.
```

(Continued)

Algorithm 1 (continued)

```
    3.2 Discriminator Training:
        (a) Train the GAN discriminator to differentiate between normal and fall sequences.
Step 4: Object Detection with YOLOv8
    4.1 Training YOLOv8:
        (a) Fine-tune YOLOv8 on the prepared dataset for detecting and locating people in frames.
        (b) Integrate Spatial Attention:
            i. Apply Spatial Attention modules to YOLOv8 to enhance focus on key spatial regions,
               improving detection accuracy.
    4.2 Real-time Detection:
        (a) Deploy YOLOv8 for real-time object detection in video streams.
Step 5: Fall Detection Algorithm
    5.1 Shape Change Monitoring:
        (a) Approximate the detected person using simple shapes (e.g., ellipses, rectangles). Monitor
            aspect ratio changes over consecutive frames to detect falls.
    5.2 Pose Estimation:
        (a) Extract and analyze key points to determine the posture. Track movements of joints (e.g.,
            head, hips, knees) to detect abnormal postures indicative of falls.
    5.3 Inactivity Detection:
        (a) Monitor prolonged inactivity following a fall to confirm detection.
```

After Pose Estimation is performed in Step 5, the system extracts keypoints representing the spatial (x, y) and temporal (z) coordinates of 15 major human joints across a sequence of frames. These keypoints are fed into a CNN-based fall detection model, specifically trained to classify falls and non-fall activities based on the dynamic changes in keypoint positions. The CNN model identifies patterns in the keypoints, such as abrupt downward movements or unusual body postures, that indicate a fall. This process allows for the accurate classification of fall events in real time. The fall detection model is lightweight and optimized to work efficiently on the Xilinx Kria™ K26 System-on-Module (SOM), ensuring compatibility with edge computing environments.

The proposed model differentiates falls from lying-down actions by analyzing the temporal and kinematic characteristics of the extracted keypoints (x, y, z). Falls are characterized by rapid, abrupt downward movements of keypoints, particularly the head, torso, and legs, over a short duration. These sudden changes in posture are distinct from the smooth and controlled transitions observed in lying-down actions. The CNN-based fall detection model processes the sequence of keypoint movements to learn these temporal and spatial differences, enabling it to classify actions accurately. The training dataset includes real and GAN-augmented examples of falls and lying-down actions to enhance further the model's ability to differentiate these activities. This approach improves the robustness of the model by exposing it to a wide variety of scenarios and movement patterns during training.

Table 1 summarizes the specifications of the preparation stage.

Table 1: Specifications of preparation stage

Item	Specification(s)
Dataset Data types	DiverseFALL10500 and CAUCAFall Videos

(Continued)

Table 1 (continued)

Item	Specification(s)
Activities covered Environments	Normal activities, Fall incidents Various (indoor, outdoor, etc.)
Lighting conditions	Diverse
Perspectives	Multiple
Normalization Resizing	Normalize video frames Resize to a standard size (224x224)
Annotation	Bounding boxes around people, labels (fall/non-fall)

Once the data is prepared, the next step is feature extraction using Faster R-CNN, as summarized in Algorithm 2. Faster R-CNN is a robust model for object detection that provides detailed feature extraction capabilities. It identifies bounding boxes around detected objects (people in this case) and extracts fine-grained features related to posture and movement. Channel Attention modules are integrated after each convolutional layer in the Faster R-CNN to enhance these features. Channel Attention helps focus on the most informative channels of the feature maps, effectively amplifying relevant features and suppressing less important ones. Additionally, a pose estimation model, such as OpenPose, is applied to the outputs of Faster R-CNN to extract the detected person's key points (joints). These key points provide detailed information about the person's posture and movements, essential for accurately detecting falls. Table 2 summarizes the specifications of the feature extraction stage.

Algorithm 2: Feature extraction with Faster R-CNN

```
Input: DiverseFALL10500 and CAUCAFall datasets
Output: Enhanced features and key points
1. Load pre-trained Faster R-CNN model
    faster_rcnn ← load_model("FasterRCNN", weights="imagenet")
2. Function APPLY_CHANNEL_ATTENTION(feature_map)
    return ChannelAttention(feature_map)
end function
3. Function POSE_ESTIMATION(bounding_box)
    keypoints ← OpenPose.detect_keypoints(bounding_box)
    return keypoints
end function
4. for each frame in dataset do
    bounding_boxes, scores, classes ← faster_rcnn.detect(frame)
    enhanced_features ← []
    for each bbox in bounding_boxes do
        feature_map ← extract_feature_map(faster_rcnn, bbox)
        enhanced_feature_map ← apply_channel_attention(feature_map)
        enhanced_features.append(enhanced_feature_map)
    end for
```

(Continued)

Algorithm 2 (continued)

```
    keypoints ← []
    for each bbox in bounding_boxes **do**
        keypoint ← pose_estimation(bbox)
        keypoints.append(keypoint)
    end for
    save_features_and_keypoints(enhanced_features, keypoints)
**end for**
```

Table 2: Specifications of feature extraction stage

Item	Specification(s)
Model	Faster R-CNN
Feature extraction	Detailed feature extraction on bounding boxes Applied after each convolutional layer
Channel attention	
Key point detection	Pose estimation model (e.g., OpenPose) Key points (joints) of detected person
Output	

GANs are then employed for data synthesis and enhancement. Synthetic data generation involves training the GAN to produce realistic sequences of falls, which are added to the training dataset. This augmentation helps improve the robustness of the model by exposing it to various fall scenarios. Squeeze-and-excitation blocks are integrated within the GAN architecture to enhance synthetic data quality further. These blocks improve the ability of the generator to create realistic fall scenarios by adaptively recalibrating channel-wise feature responses. The discriminator in the GAN is trained to differentiate between regular and fall sequences, refining the ability of the model to detect falls by learning from accurate and generated data. This step ensures the model can handle various fall types and environments, enhancing its generalization capabilities.

The GAN is employed after feature extraction to augment the keypoint-based feature space (x, y, z) with realistic variations that enhance the robustness of the fall detection model. By operating on the extracted keypoints, the GAN generates task-specific augmentations, such as variations in joint trajectories, simulated noise, and occlusion scenarios, which are directly relevant to fall-related dynamics. This approach avoids the computational complexity and irrelevance of augmenting raw data while ensuring efficient and meaningful diversity in the training dataset. The GAN operates solely during the training phase to enrich the dataset, enabling the fall detection model to generalize to real-world scenarios and improve its robustness to environmental variations and sensor noise.

Table 3 summarizes the specification of GAN, and Algorithm 3 illustrates the GANs for data synthesis and enhancement.

Table 3: Specifications of GAN

Item	Specification(s)
Synthetic data generation	GANs
GAN architecture	Includes squeeze-and-excitation blocks Realistic fall scenarios generated
Training data	by GAN
Discriminator training	Differentiate between normal and fall sequences

Algorithm 3: GANs for data synthesis and enhancement

```
Input: Real fall images
Output: Enhanced synthetic fall sequences
Function BUILD_GENERATOR
    Define and compile GAN generator model with Squeeze-and-Excitation blocks
    return generator model
end function
Function BUILD_DISCRIMINATOR
    Define and compile GAN discriminator model
    return discriminator model
end function
Function TRAIN_GAN(epochs, batch_size=128, save_interval=50)
    Load and preprocess real fall images
    for each epoch from 1 to epochs do
        Train discriminator on real and synthetic images
        Train generator through combined model
        if epoch % save_interval == 0 then
            Save generated images
        end if
end for
end function
Initialize and compile GAN models
Train the GAN
```

YOLOv8 is then fine-tuned on the prepared dataset for object detection. YOLOv8 significantly improved over its predecessors due to its enhanced speed and accuracy, making it well-suited for real-time applications. The model's architecture includes improved feature extraction layers and optimized training processes. Spatial Attention modules are applied to enhance the performance of YOLOv8. These modules help the model focus on important regions in the spatial domain, improving detection accuracy by highlighting areas most likely to contain relevant objects. Once trained, YOLOv8 is implemented for real-time object detection in video streams, ensuring it can quickly and accurately process frames.

Integrating the Spatial Attention mechanism into YOLOv8 enhances its feature extraction capabilities by focusing on the most relevant regions in the spatial domain, thus improving the model's ability to identify and concentrate on important parts of the image. This mechanism generates a spatial attention map highlighting significant areas, allowing the model to prioritize these regions more heavily in the feature maps. As a result, the enhanced feature maps become more informative, leading to better object detection performance by reducing false positives and negatives and increasing overall detection accuracy. This improvement makes YOLOv8 more effective in complex and varied environments, such as detecting human falls in the DiverseFALL10500 and CAUCAFall dataset. Table 4 summarizes the specifications of improved YOLOv8. Algorithm 4 depicts the object detection with YOLOv8.

Table 4: Specification of improved YOLOv8

Item	Specification(s)
Model	YOLOv8
Training	Fine-tune on DiverseFALL10500 and CAUCAFall dataset
Spatial attention	Applied to enhance detection accuracy
Real-time detection	Implement YOLOv8 in video streams

Algorithm 4: Object detection with YOLOv8

```
Input: Pre-trained YOLOv8 model, DiverseFALL10500 and CAUCAFall dataset, video stream
Output: Real-time object detection
Procedure OBJECTDETECTIONWITHYOLOv8
    Load pre-trained YOLOv8 model
    Fine-tune YOLOv8 on DiverseFALL10500 and CAUCAFall dataset
    for each epoch from 1 to num_epochs do
        for each batch in dataset do
            Extract images and labels from batch
            Perform forward pass through YOLOv8
            Apply Spatial Attention to feature maps
            Compute loss and backpropagate
            Update model weights
        end for
    end for
    Initialize video stream
    while video stream is open do
        Capture frame from video stream
        if frame is valid then
            Detect objects using YOLOv8
            Apply Spatial Attention to detections
            Draw bounding boxes on the frame
            Display the frame
        else
            Break
        end if
    end while
    Release video stream and close all windows
end procedure
```

The final fall detection algorithm includes shape change monitoring, posture analysis, and inactivity detection. Shape change monitoring involves approximating the detected person with simple shapes and monitoring aspect ratio changes over consecutive frames. Posture analysis uses the extracted key points to track movements indicative of a fall, while inactivity detection identifies prolonged inactivity periods after a fall, serving as an additional verification step. This comprehensive approach ensures reliable and accurate fall detection in various scenarios. Table 5 illustrates the details of the fall detection algorithm.

Table 5: Fall detection algorithm

Item	Description	Technique
Shape change monitoring	Approximate detected person with simple shapes and monitor aspect ratios	Geometric shape approximation
Posture analysis	Analyze a person's posture using extracted key points	Pose estimation, key point tracking
Inactivity detection	Identify prolonged periods of inactivity after a fall	Time-based analysis

Table 6 summarizes the specifications and techniques for each methodology stage, facilitating implementation in Python.

Table 6: Summary of techniques and libraries

Library/Tool	Usage
OpenCV	Video frame normalization, resizing
NumPy	Data manipulation
PIL	Image processing
LabelImg, CVAT	Data annotation
TensorFlow, PyTorch	Model implementation, training
SE-ResNet, CBAM	Channel attention integration
OpenPose, DeepPose	Pose estimation
GANs	Synthetic data generation
Squeeze-and-excitation networks	Data enhancement
Adam optimizer	Model training optimization
Spatial attention mechanism	Detection accuracy improvement

In our experiments, we deployed the proposed R-CNN_GAN_YOLOv8 model on the Xilinx Kria™ K26 System-on-Module (SOM), which features a quad-core Arm® Cortex®-A53 processor and a programmable logic fabric optimized for AI workloads. The hardware environment included an Intel® Core™ i7-10700 CPU for training and testing, alongside an NVIDIA RTX 2080 Ti GPU with 11GB GDDR6 memory for model training. For edge deployment, we utilized the Xilinx Kria™ K26 SOM, optimising inference times for real-time applications.

4 Results and Discussion

In this section, we present the results of our methodology for detecting human falls using GANs, Faster R-CNN, and improved YOLOv8 with Spatial Attention. The results are analyzed and discussed across three key subsections. First, Evaluation Metrics outline the metrics used to assess the performance of our models, including precision, recall, F1-score, and computational efficiency. Next, Performance Evaluation provides a detailed analysis of the model's performance on the DiverseFALL10500 and CAUCAFall datasets, highlighting the improvements achieved by integrating attention mechanisms. Finally, in the Discussion, we interpret the results, compare them with existing approaches, and explore the implications of our findings for real-world applications and future research directions. Fig. 7 shows a fall detection snapshot using the proposed approach.

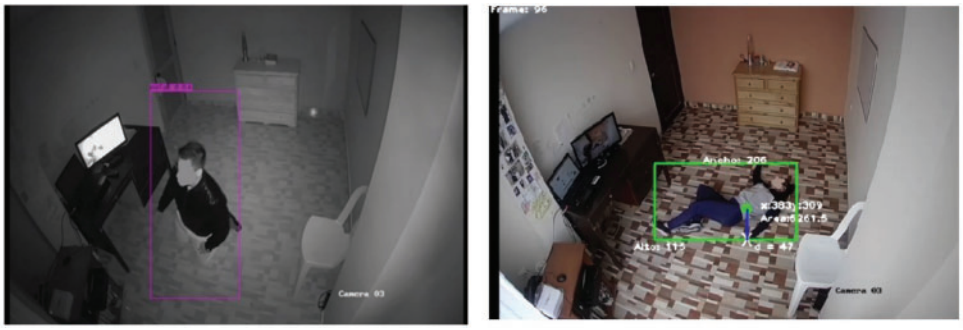


Figure 7: Fall detection using the proposed detection approach

4.1 Evaluation Metrics

This study employed several commonly used evaluation metrics for object detection, including Precision, Recall, mean Average Precision (mAP), and the number of Parameters. True Positive (TP) denotes instances where the classifier correctly identifies a positive sample, while True Negative (TN) indicates cases where a negative sample is correctly classified. False Positive (FP) and False Negative (FN) represent the number of false alarms and missed detections, respectively. These metrics comprehensively assess the model's performance, capturing accuracy and computational efficiency.

Precision, which is computed as shown in Eq. (1).

$$P = \frac{TP}{TP + FP} \tag{1}$$

Recall, which is computed as shown in Eq. (2).

$$P = \frac{TP}{TP + FN} \tag{2}$$

Average Precision (AP) is a metric used to assess the performance of single-class object detection models. It represents the area under the Precision-Recall curve, where Precision indicates the accuracy of optimistic predictions, and Recall indicates the proportion of true positives correctly identified. A higher AP value signifies better model effectiveness in detecting objects accurately.

Mean Average Precision (mAP), which is computed as shown in Eq. (3).

$$\text{mAp } 50 = \frac{1}{\overrightarrow{N}} \sum_{\text{i}=1}^{\text{N}} \text{AP}_{\text{i}}^{50} \tag{3}$$

$$\text{mAp } 75 = \frac{1}{\overrightarrow{N}} \sum_{\text{i}=1}^{\text{N}} \text{AP}_{\text{i}}^{75} \tag{4}$$

where:

- N: Total number of classes or object categories.
- AP^{50}: Average precision for a specific class at the IoU threshold of 0.5.
- AP^{75}: Average precision for a specific class at the IoU threshold of 0.75.

The experimental setup for evaluating the proposed model and its variants is summarized in Table 7, detailing the hardware and software configurations, datasets, training parameters, and evaluation metrics used to ensure reproducibility and robust performance assessment.

Table 7: Experimental setup for model evaluation

Component	Details
Hardware configuration	**Training:** Intel Core i7-10700 CPU, NVIDIA RTX 2080 Ti GPU (11 GB), 32 GB RAM **Deployment:** Xilinx Kria™ K26 SOM with quad-core Arm® Cortex®-A53 processor
Software environment	Ubuntu 20.04, Python 3.8, PyTorch 1.10
Datasets used	**CAUCAFall** for training and testing, **DIVERSEFALL** for additional evaluation **Dataset split:** 70% training, 15% validation, 15% testing
Evaluation metrics	mAP@0.5, mAP@0.75, Separate fall and no-fall met-rics
Training parameters	**Optimizer**: Adam, **Learning rate:** 0.001 (decayed by 0.1 after 10 epochs) **Batch size:** 32, **Epochs:** 200 (with early stopping) **Confidence threshold:** 0.5
Post-processing	GAN-based data augmentation for diverse keypoint generation
Evaluation setup	Real-time evaluation on Xilinx Kria™ K26 SOM Inference times recorded for fall and no-fall classifications

4.2 Datasets

Table 8 compares datasets utilized for human fall detection, highlighting the comprehensive features that make them suitable for this study. The CAUCAFall dataset [50] was selected due to its diversity in lighting conditions, occlusion scenarios, and realistic participant interactions, providing a robust foundation for model training. The DiverseFALL10500 dataset [51] was chosen for its superior variability, including high levels of inclusion in visual appearances, fall angles, age diversity, and background conditions. By combining these datasets, the study ensures that the proposed model is trained and validated on a wide range of scenarios, enabling enhanced generalization, robustness, and real-world applicability.

Table 8: Datasets. The sign ✓ shows the incorporated features while the ✓✓ presents the high level of inclusion.

Dataset	Variable visual appearances	Occlusion	Variety in fall angles	Light conditions	Age diversity	Clothing and attire variations	Background diversity
CAUCAFall [50]	–	✓	✓	natural, artificial, no light	–	–	✓
DiverseFALL10500 [51]	✓✓	✓✓	✓✓	natural, artificial, night	✓✓	✓✓	✓✓

4.3 Inference Times for Fall and No-Fall

Table 9 presents the proposed system's inference times for fall and no-fall classifications. The average inference time per frame was 43 ms for fall scenarios and 39 ms for no-fall scenarios. These results demonstrate the system's capability to operate within real-time constraints, supported by the efficiency of the Xilinx Kria™ K26 SOM hardware.

Table 9: Inference times for fall and no-fall classifications

Scenario	Average inference time (ms)	Hardware platform
Fall detection	43	Xilinx Kria™ K26 SOM
No-fall classification	39	Xilinx Kria™ K26 SOM

The observed inference times highlight the computational efficiency of the proposed system. By leveraging hardware acceleration on the Xilinx Kria™ K26 SOM, the system achieves real-time performance suitable for practical deployment in real-world environments. This demonstrates the viability of the proposed approach for fall detection and classification tasks.

4.4 Performance Evaluation

Fig. 8 shows the training and validation loss for the Faster RCN_GAN_YOLOv8 model over 200 epochs, showing a clear and smooth convergence, with both loss curves decreasing steadily. Initially, the training and validation losses start at around 0.12, indicating a significant error rate at the beginning of the training process. However, as the training progresses, the losses drop rapidly, particularly in the first 50 epochs, before stabilizing and decreasing slowly. This behaviour suggests that the model quickly learns the underlying patterns in the data and continues to fine-tune its parameters as training proceeds.

The training loss curve consistently decreases and approaches zero as the number of epochs increases, indicating that the model effectively minimises the error on the training dataset. The smooth and gradual decline in the training loss reflects the model's ability to learn from the training data without significant interruptions or overfitting. The lack of sudden spikes or plateaus in the training loss curve suggests a well-tuned learning rate and stable training process. This smooth reduction in training loss is a positive indicator of the model's learning efficiency and the effectiveness of the training strategy employed.

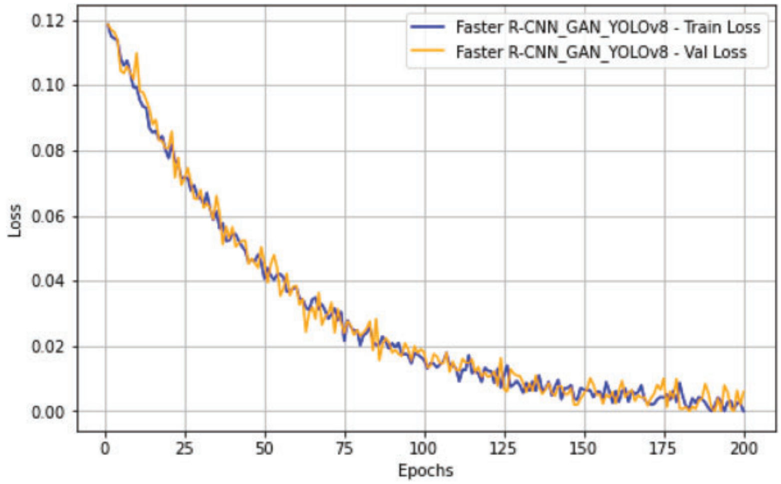


Figure 8: Training and validation loss

The validation loss curve follows a similar trajectory to the training loss, starting slightly higher but converging at a similar rate. The close alignment between the training and validation loss curves strongly indicates that the model generalizes well to unseen data with minimal overfitting. As the validation loss decreases steadily and aligns closely with the training loss, the model's performance on the validation set improves alongside its performance on the training set. This alignment suggests that the enhancements incorporated into the Faster R-CNN_GAN_YOLOv8 model, such as using GANs for data augmentation and attention mechanisms for better feature extraction, contribute to robust generalization and overall model effectiveness.

Table 10 provides a detailed comparison of various models for detecting falls across two datasets: DiverseFALL10500 and CAUCAFall, using metrics such as mAP (mean Average Precision), Precision, F1-score, and Recall. The results indicate that the Proposed model outperforms all others, achieving the highest mAP, Precision, F1-score, and Recall on both datasets. Specifically, the Proposed model records an mAP of 0.9507 on DiverseFALL10500 and 0.996 on CAUCAFall, with near-perfect Precision and Recall on the latter, highlighting its exceptional detection capabilities. Performance on the DiverseFALL10500 dataset shows that YOLOv8s, YOLOv8+SE, and YOLOv8+CA are also strong contenders, with mAP values of 0.920, 0.922, and 0.944, respectively. Models enhanced with attention mechanisms, such as YOLOv8+SE and YOLOv8+CA, consistently outperform baseline models, demonstrating the advantages of incorporating advanced feature extraction techniques. This improvement is particularly evident in their Precision and Recall values, which reflect enhanced detection accuracy and reliability.

The CAUCAFall dataset exhibits generally higher performance metrics across all models than DiverseFALL10500. This suggests the CAUCAFall dataset may be less complex or better aligned with the models' training data. Models like YOLOv8+Shuffle Attention and YOLOv8+SimAM also achieve near-perfect scores, indicating their robustness in handling this dataset. Notably, the Proposed model excels on CAUCAFall, achieving an mAP of 0.996 and a Recall of 0.9993, making it highly effective for real-world fall detection applications. The inclusion of attention mechanisms plays a critical role in enhancing performance. Models such as YOLOv8+SE, YOLOv8+CBAM, and YOLOv8+Shuffle Attention outperform their counterparts without these mechanisms, confirming the value of attention modules in capturing subtle fall patterns. Lightweight models like YOLOv8-tiny also perform well, with an mAP of 0.9003 on DiverseFALL10500 and

0.9955 on CAUCAFall, making them suitable for resource-constrained environments despite their slightly lower overall accuracy.

Table 10: Comparison with SOTA

S. No.	Model	DiverseFALL10500				CAUCAFall			
		mAP	Precision	F1-score	Recall	mAP	Precision	F1-score	Recall
1	Faster R-CNN	0.831	0.837	0.813	0.809	0.9903	0.9891	0.9902	0.9924
2	YOLOv3	0.842	0.848	0.827	0.809	0.9912	0.9904	0.9916	0.9931
3	YOLOv4	0.825	0.818	0.801	0.794	0.9896	0.9901	0.9898	0.9913
4	YOLOv5n	0.870	0.810	0.839	0.852	0.9924	0.9921	0.9935	0.9942
5	YOLOv5s	0.906	0.895	0.878	0.845	0.9932	0.9926	0.9954	0.9961
6	YOLOv5m	0.848	0.850	0.825	0.807	0.9911	0.9914	0.9923	0.9927
7	YOLOv5l	0.835	0.833	0.813	0.837	0.9926	0.9922	0.9934	0.9936
8	YOLOv5x	0.834	0.769	0.797	0.829	0.9898	0.9892	0.9902	0.9906
9	YOLOv8n	0.863	0.810	0.822	0.837	0.9925	0.9921	0.9923	0.9935
10	YOLOv8s	0.920	0.900	0.895	0.859	0.9950	0.9948	0.9952	0.9950
11	YOLOv8m	0.899	0.880	0.868	0.851	0.9949	0.9948	0.9952	0.9950
12	YOLOv8l	0.855	0.856	0.846	0.845	0.9947	0.9914	0.9934	0.9953
13	YOLOv8x	0.903	0.885	0.874	0.853	0.9945	0.9937	0.9941	0.9953
14	YOLOv8-tiny [52]	0.9003	0.8703	0.8661	0.862	0.9955	0.9940	0.9951	0.9956
15	YOLOv8+SE [53]	0.922	0.8811	0.8956	0.9107	0.9962	0.9956	0.9964	0.9968
16	YOLOv8+CBAM [54]	0.915	0.893	0.8792	0.866	0.9958	0.9950	0.9960	0.9965
17	YOLOv8+CA [55]	0.944	0.905	0.8750	0.847	0.9966	0.9960	0.9968	0.9970
18	YOLOv8+SimAM [53]	0.9271	0.8999	0.90	0.9021	0.9965	0.9962	0.9966	0.9971
19	YOLOv8+Shuffle Attention [56]	0.9342	0.9312	0.9036	0.8777	0.9967	0.9965	0.9970	0.9973
20	FE-YOLO [57]	0.9132	0.889	0.8949	0.901	0.9959	0.9954	0.9963	0.9967
21	YOLOV8S [51]	0.935	0.900	0.884	0.859	0.9954	0.9982	0.9981	0.9984
22	**Proposed**	**0.9507**	**0.9307**	**0.9298**	**0.929**	**0.996**	**0.9972**	**0.9982**	**0.9993**

The proposed model outperforms State-of-the-Art (SOTA) models due to its innovative architecture and enhancements that address critical limitations in existing fall detection systems. By integrating advanced components such as YOLOv8, Faster R-CNN, and GANs, the model effectively combines real-time detection, detailed feature extraction, and data augmentation. YOLOv8 contributes to fast and accurate object detection, while Faster R-CNN ensures precise feature extraction by capturing subtle postural and movement cues. Ad- ditionally, GANs enhance the dataset diversity by generating realistic synthetic fall scenarios, addressing the issue of insufficient training data. A key factor in the model's success is incorporating attention mechanisms. Channel attention, implemented in Faster R-CNN, improves feature selection by focusing on the most relevant channels. In contrast, spatial attention in YOLOv8 enhances the detection of critical regions in the spatial domain. Furthermore, squeeze-and-excitation blocks in GANs recalibrate feature responses, ensuring that the generated synthetic data closely resembles real-world scenarios. These attention mechanisms collectively enhance the model's ability to detect falls accurately, even in challenging environments.

The robustness of the proposed model is further reinforced by its training on diverse datasets, including DiverseFALL10500 and CAUCAFall. These datasets include various scenarios, such as varied lighting conditions, different fall angles, and diverse participant demographics. By balancing real and GAN-generated data, the model demonstrates excellent generalization capabilities, ensuring reliable performance in real-world applications. This approach is particularly practical in handling complex scenarios, such as occlusions and low-contrast environments, which often challenge conventional models. The proposed model's holistic design also contributes to its superior performance. Integrating real-time detection capabilities from YOLOv8 with the precision of Faster R-CNN and the data augmentation capabilities of GANs creates a comprehensive solution. This synergy addresses multiple facets of the fall detection problem, from accurate detection to scalability and deployment readiness. The system's optimization for edge devices ensures efficient operation, making it practical for home monitoring or assisted living facilities applications. In conclusion, the proposed model achieves the highest mAP, Precision, F1-score, and Recall across both DiverseFALL10500 and CAUCAFall datasets, setting a new benchmark in human fall detection. Its innovative architecture, advanced feature extraction, robust data augmentation, and real-time performance collectively enhance its accuracy, reliability, and adaptability. These advancements make the proposed model a significant step forward in addressing the challenges of fall detection in diverse and uncontrolled environments.

4.5 Discussion and Future Works

The proposed approach, which integrates R-CNN, GANs, and an improved YOLOv8, demonstrates significant advancements in human fall detection systems. Including GANs for data synthesis significantly enhances the robustness and diversity of the training dataset. By generating realistic fall scenarios, the GAN component ensures the model is exposed to various fall types and conditions, improving its ability to generalize to real-world scenarios. This enhancement is particularly crucial given the limitations of traditional datasets in capturing the full spectrum of potential fall events. Integrating Faster R-CNN for feature extraction brings detailed and nuanced human posture and movement analysis. By utilizing Channel Attention mechanisms within the Faster R-CNN architecture, the model can focus on the most informative features within the data. This attention mechanism effectively amplifies relevant features while suppressing less important ones, leading to more accurate detection of falls. The detailed feature extraction provided by Faster R-CNN ensures that even subtle cues indicative of falls are captured and analyzed, contributing to the overall precision and recall of the model.

YOLOv8, with its inherent speed and accuracy improvements, is further enhanced by integrating Spatial Attention mechanisms. These mechanisms allow the model to focus on significant regions within the spatial domain, thus improving detection accuracy. The Spatial Attention modules ensure that the model gives more weight to areas most likely to contain relevant objects, which is critical for accurately identifying falls in complex and varied environments. Using YOLOv8 ensures the system can operate in real-time, providing timely alerts during a fall. The results of this study indicate that the proposed Faster R-CNN_GAN_YOLOv8 model outperforms existing state-of-the-art models.

The high precision and recall rates indicate that the model is highly reliable in detecting falls, which is crucial for applications in environments where elderly individuals or those with mobility challenges are present. The low rate of false positives minimises unnecessary alarms, reducing the likelihood of unnecessary interventions. Similarly, the low rate of false negatives ensures that actual fall incidents are promptly detected, enabling timely medical responses and potentially reducing the severity of injuries. In conclusion, the proposed approach significantly contributes to fall detection systems. The model achieves superior detection accuracy and computational efficiency by leveraging the strengths of Faster R-CNN, GANs, and YOLOv8 and incorporating advanced attention mechanisms. These advancements are essential for developing reliable and effective fall detection systems, particularly in real-world, uncontrolled environments. The proposed model offers a robust solution that can enhance safety and provide timely assistance, ultimately the quality of life for individuals at risk of falls.

4.5.1 Computational Efficiency

While R-CNNs are computationally intensive, our implementation mitigates these challenges through hardware acceleration on the Xilinx Kria™ K26 SOM and the integration of quantization techniques. These optimizations significantly reduce the computational load, allowing the system to achieve real-time performance suitable for deployment in practical environments. Quantization, in particular, reduces the model size and computational complexity without significantly compromising accuracy, making it an effective approach for edge devices. Despite these optimizations, there remains scope for further improvement. Future work will explore lightweight variants of R-CNN, such as Fast R-CNN and Faster R-CNN, designed to address computational bottlenecks. Moreover, alternative architectures such as Single Shot MultiBox Detector (SSD) and MobileNet are promising candidates for improving computational efficiency while maintaining robust performance in fall detection tasks. These models have been shown to perform well in real-time scenarios with limited computational resources, making them suitable for edge-based applications.

The exploration of lightweight models aims to improve computational efficiency and extends the system's applicability to diverse deployment environments. For instance, Assisted Living Facilities (ALFs) often require large-scale coverage, which may require multiple devices. Deploying computationally efficient models in such settings could help reduce hardware costs, power consumption, and network bandwidth requirements, enhancing the system's scalability. In addition to lightweight models, distributed edge computing frameworks could improve system efficiency. Distributing computational tasks across multiple edge devices minimises latency and bottlenecks associated with centralized processing. This approach also reduces the dependency on high-bandwidth network infrastructure, making the system more resilient and adaptable to different environments.

Furthermore, evaluating the proposed solutions through rigorous real-world testing in diverse scenarios is essential. Pilot studies in ALFs, private homes, and resource-constrained settings will provide valuable insights into the system's performance, scalability, and user acceptance. Feedback from stakeholders, including caregivers and facility managers, will help refine the system to meet practical needs more effectively. In conclusion, while our current implementation addresses some computational challenges through hardware acceleration and quantization, future work must focus on adopting lightweight architectures, distributed edge computing frameworks, and adaptive inference techniques. These advancements can potentially improve computational efficiency, scalability, and usability, enabling the deployment of fall detection systems in a wider range of real-world settings.

4.5.2 Addressing Privacy Concerns in Video Based Fall Detection Systems

While video-based fall detection systems offer significant potential for improving the safety and well-being of elderly individuals, privacy concerns remain a critical barrier to their widespread adoption. These concerns are particularly acute in sensitive areas of the home, such as bedrooms and bathrooms, where most falls occur [58]. The practicality of deploying RGB cameras in such private spaces is often limited due to societal and cultural norms, underscoring the need for privacy-centric solutions.

The proposed system adopts a privacy-preserving approach by processing RGB data only to extract bounding boxes of individuals and discarding the RGB frames immediately thereafter. This ensures that no identifiable visual information is stored or transmitted, with subsequent analysis relying solely on depth data and joint coordinates (x, y, z). While this method significantly mitigates privacy risks, additional efforts are necessary to enhance the system's acceptability and adoption further.

Future work will explore integrating alternative sensor technologies to complement the current vision-based approach. Infrared cameras, for instance, can capture movement without revealing visual details, making them suitable for deployment in sensitive areas. Wearable devices, such as accelerometers and gyroscopes, offer another promising solution for continuous monitoring without compromising privacy. Combining these technologies with depth-based video systems can create a hybrid framework that ensures comprehensive coverage while maintaining user privacy. Additionally, future research will address user trust and perceptions, pivotal for successfully adopting privacy-centric systems. Educating users about the system's privacy-preserving features and providing transparency in data processing and storage practices will be key to building trust. This includes clear communication about how sensitive data is handled and the measures to ensure confidentiality.

Finally, large-scale field trials in real-world environments will be conducted to evaluate the practicality of these privacy-focused approaches. These trials will assess not only the technical performance of the system but also its acceptability and usability among end-users. Feedback from these trials will inform iterative improvements to enhance functionality and user satisfaction. Future efforts will address video-based fall detection systems' privacy and practicality challenges. By integrating alternative sensor technologies, educating users, and conducting extensive field testing, we seek to develop a comprehensive and privacy-respecting solution that facilitates the adoption of fall detection systems in diverse real-world settings.

4.5.3 Handling Multiple Persons in a Single Frame

One of the challenges in fall detection systems is managing scenarios where multiple persons appear in a single frame [56]. The proposed architecture addresses this challenge by employing a systematic approach that ensures the independent analysis of each individual in the frame. The YOLOX model, used in the Pose Estimation step, detects all individuals in the frame and generates bounding boxes for each person.

This step ensures that each individual is isolated and processed separately, enabling the system to manage multiple persons effectively. The HPE model extracts the keypoints (x, y, z) of 15 major human joints for each detected bounding box. This independent extraction allows the system to evaluate the pose and motion of each individual without interference from others in the same frame.

The extracted keypoints for each person are then fed into the CNN-based fall detection model, which independently classifies the activity of each individual as either a fall or a non-fall. The classification is based on the dynamic changes in the keypoint positions, such as sudden downward movements or unusual body postures indicative of a fall. The architecture is optimized for real-time processing on edge devices, specifically the Xilinx Kria™ K26 System-on-Module (SOM). Hardware acceleration and an efficient processing pipeline ensure scalability, allowing the system to handle multiple individuals simultaneously while maintaining real-time performance. This capability makes the proposed architecture suitable for environments such as Assisted Living Facilities (ALFs) or crowded settings where various persons may need to be monitored concurrently.

The proposed system overcomes the potential challenges associated with multi-person scenarios by isolating and analysing each individual independently. This ensures accurate fall detection in real-world applications where multiple individuals may be in the same frame. Future work will optimise the system for complex multi-person interactions, such as overlapping movements or occlusions.

4.5.4 Scalability Challenges in Video-Based Fall Detection Systems

Scalability presents a significant challenge for deploying video-based fall detection systems, especially in environments such as Assisted Living Facilities (ALFs), where elderly individuals frequently traverse and congregate in various areas [59]. The comprehensive coverage required in such settings necessitates the deployment of cameras in all relevant spaces. Furthermore, occlusions caused by furniture or other obstacles may obstruct a camera's view of human joints, affecting the system's ability to detect falls reliably. Addressing these occlusions often requires multiple cameras per room, increasing the infrastructure requirements for hardware, installation, power supply, and network connectivity. The cost implications of such deployments are substantial, as they involve expenses for additional cameras, mounting hardware, power supply systems, and WiFi infrastructure. These costs may pose a barrier to the practical adoption of the proposed system, particularly in resource-constrained environments. Moreover, managing and maintaining a large-scale camera network in ALFs further complicates scalability.

To address these challenges, future system iterations should explore hybrid solutions that reduce reliance on extensive camera networks while maintaining comprehensive monitoring capabilities. One potential avenue is the integration of wearable devices, such as accelerometers or gyroscopes, to complement video-based systems [60]. Wearable sensors can continuously monitor individuals and reduce the need for multiple cameras in private or obstructed areas. Additionally, environmental sensors, such as infrared or ultrasonic detectors, may be cost-effective alternatives in specific zones where video coverage is less practical.

Another promising direction is the development of advanced multi-camera coordination algorithms. Such algorithms could optimize the coverage provided by multiple cameras, ensuring that obstructions are minimized while reducing the overall number of devices required. By leveraging techniques such as spatial stitching and depth fusion, these algorithms can combine data from multiple cameras to generate a comprehensive and seamless view of the monitored environment. This approach not only enhances system performance but also reduces hardware redundancy. Furthermore, implementing distributed edge computing can play a critical role in improving scalability. By processing data locally at the camera level, edge computing minimizes the bandwidth required for transmitting data to centralized servers and reduces overall network congestion. This localized processing also allows for faster decision-making, making the system more suitable for real-time fall detection in large-scale environments.

Pilot studies in ALFs should be conducted to evaluate the cost-benefit trade-offs and assess the system's scalability in real-world settings. These studies can provide insights into the infrastructure requirements, user acceptance, and overall feasibility of large-scale deployments. Feedback from stakeholders, including facility managers and caregivers, will be invaluable in refining the system to meet the unique demands of such environments. In conclusion, while scalability poses a significant challenge for video-based fall detection systems, integrating hybrid solutions, leveraging advanced algorithms, and employing distributed computing techniques offer promising pathways to address these issues. Future research should prioritize these directions to ensure the system's feasibility, affordability, and reliability in large-scale applications like ALFs.

5 Conclusion

This paper presents a novel fall detection model that outperforms state-of-the-art methods by addressing key challenges in real-world applications. The proposed model integrates advanced deep learning architectures, including YOLOv8 for real-time detection, Faster R-CNN for precise feature extraction, and GANs for enhancing dataset diversity. Additionally, incorporating attention mechanisms such as channel attention, spatial attention, and squeeze-and-excitation modules further refine the model's ability to detect falls accurately in complex and diverse environments. The model was evaluated on two datasets, DiverseFALL10500 and CAUCAFall, and achieved the highest performance across all key metrics, including mAP, Precision, F1-score, and Recall. These results demonstrate the robustness and reliability of the proposed system, particularly in challenging scenarios such as occlusions, low-contrast settings, and varied fall conditions. The model's compatibility with edge devices and ability to handle diverse input data also highlight its practical applicability for real-time monitoring in settings such as homes, hospitals, and assisted living facilities.

The findings underscore the importance of combining real-time detection capabilities, advanced feature extraction techniques, and robust data augmentation to create a holistic fall detection system. This research contributes significantly to advancing fall detection technologies by addressing the critical limitations of existing methods. Future work will explore further optimization of the model for deployment in resource-constrained environments and its integration with broader healthcare monitoring systems for enhanced safety and usability.

Acknowledgement: The authors express their gratitude to the University of Sfax, Tunisia, for administrative and technical support.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Study Conception and Design: Nizar Zaghden and Emad Ibrahim; Data Collection: Mukaram Safaldin and Mahmoud Mejdoub; Analysis and Interpretation of Results: Nizar Zaghden, Emad Ibrahim, Mukaram Safaldin and Mahmoud Mejdoub; Draft Manuscript Preparation: Emad Ibrahim and Mukaram Safaldin. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Daramola O, Awunor N, Akande T. The challenges of retirees and older persons in Nigeria; a need for close attention and urgent action. Int J Trop Dis Health. 2019;34(4):1–8. doi:10.9734/ijtdh/2018/v34i430099.
2. You D, Hug L, Anthony D. Generation 2030/Africa. 3. United Nations Plaza, New York: UNICEF; 2014.
3. Gard G. Prevention of slip and fall accidents: risk factors, methods and suggestions for prevention. Phys Therapy Rev. 2000;5(3):175–82. doi:10.1179/ptr.2000.5.3.175.
4. Lockhart TE. Fall accidents among the elderly. In: Karwowski W, editor. International encyclopedia of ergonomics and human factors. 2nd ed. CRC Press; 2007. p. 2626–30.
5. Newaz NT, Hanada E. The methods of fall detection: a literature review. Sensors. 2023;23(11):5212. doi:10.3390/s23115212.
6. Ding J, Wang Y. A WiFi-based smart home fall detection system using recurrent neural network. IEEE Trans Consum Electron. 2020;66(4):308–17. doi:10.1109/TCE.2020.3021398.
7. Zhang W, Jin T, Sun T, Huang Y, Gao X, Jeung J. Interaction design of fall detection camera in smart home care scenario. J Comput Aided Des Comput Graph. 2023;35(2):238–47.
8. Akhmetshin E, Nemtsev A, Shichiyakh R, Shakhov D, Dedkova I. Evolutionary algorithm with deep learning based fall detection on Internet of things environment. J Fusion Pract Appl. 2024;14(2):132–45. doi:10.54216/FPA.140211.
9. Yu X, Park S, Kim D, Kim E, Kim J, Kim W, et al. A practical wearable fall detection system based on tiny convolutional neural networks. Biomed Signal Process Control. 2023;86(4):105325. doi:10.1016/j.bspc.2023.105325.
10. Kulurkar P, Kumar Dixit C, Bharathi VC, Monikavishnuvarthini A, Dhakne A, Preethi P. AI based elderly fall prediction system using wearable sensors: a smart home-care technology with IOT. Meas Sens. 2023;25(1):100614. doi:10.1016/j.measen.2022.100614.
11. Inturi AR, Manikandan VM, Garrapally V. A novel vision-based fall detection scheme using keypoints of human skeleton with long short-term memory network. Arab J Sci Eng. 2023;48(2):1143–55. doi:10.1007/s13369-022-06684-x.

12. Durga Bhavani K, Ferni Ukrit M. Design of inception with deep convolutional neural network based fall detection and classification model. Multimed Tools Appl. 2024;83(8):23799–817. doi:10.1007/s11042-023-16476-6.
13. Alarifi A, Alwadain A. Killer heuristic optimized convolution neural network-based fall detection with wearable IoT sensor devices. Measurement. 2021;167(13):108258. doi:10.1016/j.measurement.2020.108258.
14. Jiang H, Learned-Miller E. Face detection with the faster R-CNN. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017); 2017; Washington, DC, USA: IEEE. p. 650–7.
15. Chemelil PK. Single shot multi box detector approach to autonomous vision-based pick and place robotic arm in the presence of uncertainties. Nairobi: Jomo Kenyatta University of Agriculture and Technology (JKUAT); 2021.
16. Mubashir M. A study of fall detection: review and implementation. South Yorkshire, UK: University of Sheffield; 2011.
17. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Las Vegas, NV, USA, 2016, p. 779–88.
18. Safaldin M, Zaghden N, Mejdoub M. Moving object detection based on enhanced Yolo-V2 model. In: 2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA); 2023; İstanbul, Türkiye: IEEE. p. 1–8.
19. Safaldin M, Zaghden N, Mejdoub M. An improved YOLOv8 to detect moving objects. IEEE Access. 2024;12:59782–806. doi:10.1109/ACCESS.2024.3393835.
20. Terven J, Córdova-Esparza DM, Romero-González JA. A comprehensive review of yolo architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS. Mach Learn Knowl Extrac. 2023;5(4):1680–716. doi:10.3390/make5040083.
21. Sohan M, Sai Ram T, Reddy R, Venkata C. A review on YOLOv8 and its advancements. In: International Conference on Data Intelligence and Cognitive Informatics; 2024; Singapore: Springer. p. 529–45.
22. Hussain M. YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. Machines. 2023;11(7):677. doi:10.3390/machines11070677.
23. Goodfellow IJ. On distinguishability criteria for estimating generative models. arXiv:1412.6515. 2014.
24. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. Commun ACM. 2020;63(11):139–44. doi:10.1145/3422622.
25. Sajithvariyar VV, Aswin S, Sowmya V, Soman KP, Sivanpillai R, Brown GK. Analysis of four generator architectures of c-GAN, loss function, and annotation method for epiphyte identification. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences; 2021;44: 149–53.
26. Jabbar A, Li X, Omar B. A survey on generative adversarial networks: variants, applications, and training. ACM Comput Surv. 2021;54(8):1–49.
27. Durgadevi M. Generative adversarial network (GAN): a general review on different variants of GAN and applications. In: 2021 6th International Conference on Communication and Electronics Systems (ICCES); 2021; Coimbatore, India: IEEE. p. 1–8.
28. Girshick R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Santiago, Chile, 2015. p. 1440–8.
29. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems 28 (NIPS 2015); Vancouver, BC, Canada; 2015;28: 1–9.
30. Jiang P, Ergu D, Liu F, Cai Y, Ma B. A review of Yolo algorithm developments. Procedia Comput Sci. 2022;199(11):1066–73. doi:10.1016/j.procs.2022.01.135.
31. Sumit, Shrishti Bisht, Sunita Joshi, Urvi Rana. Comprehensive review of R-CNN and its variant architectures. Int Res J Adv Eng Hub (IRJAEH). 2024;2(04):959–66. doi:10.47392/IRJAEH.2024.0134.
32. Priyadharshini G, Dolly DRJ. Comparative investigations on tomato leaf disease detection and classification using CNN, R-CNN, fast R-CNN and faster R-CNN. Vol. 1. In: 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS); 2023; IEEE. p. 1540–5.
33. Wang X, Ellul J, Azzopardi G. Elderly fall detection systems: a literature survey. Front Rob AI. 2020;7:71. doi:10.3389/frobt.2020.00071.
34. WHO. Falls; 2021. [cited 2025 Jan 15]. Available from: https://www.who.int/news-room/fact-sheets/detail/falls.

35. González-Cańete FJ, Casilari E. A feasibility study of the use of smartwatchens in wearable fall detection systems. Sensors. 2021;21(6):2254. doi:10.3390/s21062254.

36. Vishnu C, Datla R, Roy D, Babu S, Mohan CK. Human fall detection in surveillance videos using fall motion vector modeling. IEEE Sens J. 2021;21(15):17162–70. doi:10.1109/JSEN.2021.3082180.

37. Bian ZP, Hou J, Chau LP, Magnenat-Thalmann N. Fall detection based on body part tracking using a depth camera. IEEE J Biomed Health Inform. 2014;19(2):430–9. doi:10.1109/JBHI.2014.2319372.

38. Waheed SA, Khader PSA. A novel approach for smart and cost effective IoT based elderly fall detection system using Pi camera. In: 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC); 2017; Coimbatore, India: IEEE. p. 1–4.

39. Nguyen VD, Le MT, Do AD, Duong HH, Thai TD, Tran DH. An efficient camera-based surveillance for fall detection of elderly people. In: 2014 9th IEEE Conference on Industrial Electronics and Applications; 2014; Hangzhou, China: IEEE. p. 994–7.

40. Jariyavajee C, Faphatanchai A, Saeheng W, Tuntithawatchaikul C, Sirinaovakul B, Polvichai J. An improvement in fall detection system by voting strategy. In: 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC); 2019; JeJu, Republic of Korea: IEEE. p. 1–4.

41. Rougier C, Meunier J, St-Arnaud A, Rousseau J. Fall detection from human shape and motion history using video surveillance. Vol. 2. In: 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07); 2007; IEEE. p. 875–80.

42. Agrawal SC, Tripathi RK, Jalal AS. Human-fall detection from an indoor video surveillance. In: 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT); 2017; Delhi, India: IEEE. p. 1–5.

43. Foroughi H, Rezvanian A, Paziraee A. Robust fall detection using human shape and multi-class support vector machine. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing; 2008; Bhubaneswar, India: IEEE. p. 413–20.

44. Anderson D, Luke RH, Keller JM, Skubic M, Rantz M, Aud M. Linguistic summarization of video for fall detection using voxel person and fuzzy logic. Comput Vis Image Understand. 2009;113(1):80–9. doi:10.1016/j.cviu.2008.07.006.

45. Sase PS, Bhandari SH. Human fall detection using depth videos. In: 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN); 2018; Noida, Delhi-NCR, India: IEEE. p. 546–9.

46. Yan L, Du Y. Exploring trends and clusters in human posture recognition research: an analysis using CiteSpace. Sensors. 2025;25(3):632. doi:10.3390/s25030632.

47. Su MC, Liao JW, Wang PC, Wang CH. A smart ward with a fall detection system. In: 2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe); 2017; Milan, Italy: IEEE. p. 1–4.

48. Maray N, Ngu AH, Ni J, Debnath M, Wang L. Transfer learning on small datasets for improved fall detection. Sensors. 2023;23(3):1105. doi:10.3390/s23031105.

49. Ramesh SN, Sarkar M, Audette M, Paolini C. Efficient Real-time fall prediction and detection using privacy-centric vision-based human pose estimation on the Xilinx® Kria^{TM} K26 SOM. In: 2023 IEEE Biomedical Circuits and Systems Conference (BioCAS); 2023; Oronto, ON, Canada: IEEE. p. 1–5.

50. Guerrero JCE, España EM, Añasco MM, Lopera JEP. Dataset for human fall recognition in an uncon-trolled environment. Data Brief. 2022;45(6):108610. doi:10.1016/j.dib.2022.108610.

51. Khan H, Ullah I, Shabaz M, Omer MF, Usman MT, Guellil MS, et al. Visionary vigilance: optimized YOLOV8 for fallen person detection with large-scale benchmark dataset. Image Vis Comput. 2024;149(4):105195. doi:10.1016/j.imavis.2024.105195.

52. Le DTA, Nguyen H, Jang YM. An experimental demonstration of 2D2-multiple-input-multiple-output- based deep learning for optical camera communication. Appl Sci. 2024;14(3):1003. doi:10.3390/app14031003.

53. Chen C, Hy XU, Zhu XZ, Huang X, Song J, Cao YQ, et al. FDW-YOLO: an improved indoor pedestrian fall detection algorithm based on YOLOv8. Comput Eng Sci. 2024;46(8):1455.

54. Ye L, AttFallNet Lei J. Attention-Guided Fall Detection System Using Improved YOLOv8 Network. In: 2024 IEEE 17th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC); 2024; Kuala Lumpur, Malaysia: IEEE. p. 78–85.
55. Qin Y, Miao W, Qian C. A high-precision fall detection model based on dynamic convolution in complex scenes. Electronics. 2024;13(6):1141. doi:10.3390/electronics13061141.
56. Khekan AR, Aghdasi HS, Salehpoor P. Fast and high-precision human fall detection using improved YOLOv8 model. IEEE Access. 2024;13:5271–83.
57. Zhang Y, Ma H, Xie Z. Improved fall-down detection algorithm: FE-YOLO. In: 2024 36th Chinese Control and Decision Conference (CCDC); 2024; Xi'an, China: IEEE. p. 2302–6.
58. Hill AM, Etherton-Beer C, Haines TP. Tailored education for older patients to facilitate engagement in falls prevention strategies after hospital discharge—a pilot randomized controlled trial. PLoS One. 2013;8(5):e63450. doi:10.1371/journal.pone.0063450.
59. Poonsri A, Chiracharit W. Improvement of fall detection using consecutive-frame voting. In: 2018 International Workshop on Advanced Image Technology (IWAIT); 2018; Chiang Mai, Thailand: IEEE. p. 1–4.
60. De Miguel K, Brunete A, Hernando M, Gambao E. Home camera-based fall detection system for the elderly. Sensors. 2017;17(12):2864. doi:10.3390/s17122864.